Flex static code analysis

Unique rules to find Bugs, Security Hotspots, and Code Smells in your FLEX code

Secrets

ABAP

Apex

C

C++

CloudFormation

COBOL

C#

CSS

Flex

Go

HTML

Java

JavaScript

Kotlin

Kubernetes

Objective C

PHP

PL/I

PL/SQL

Python

RPG

Ruby

Scala

Swift

Terraform

Text

TypeScript

T-SQL

VB.NET

VB6

XML

All rules 76 | 🔒 Vulnerability 5 | 🐛 Bug 9 | 🛡 Security Hotspot 1 | ☢ Code Smell 61

Tags ⌄

Search by name...

Security.allowDomain(...) should only be used in a tightly focused manner

🔒 Vulnerability

The flash.system.Security.exactSettings property should never be set to false

🔒 Vulnerability

Dynamic classes should not be used

☢ Code Smell

"LocalConnection" should be configured to narrowly specify the domains with which local connections to other Flex application are allowed

🔒 Vulnerability

"default" clauses should be first or last

☢ Code Smell

Event types should be defined in metadata tags

☢ Code Smell

Event names should not be hardcoded in event listeners

☢ Code Smell

The special "star" type should not be used

☢ Code Smell

Variables of the "Object" type should not be used

☢ Code Smell

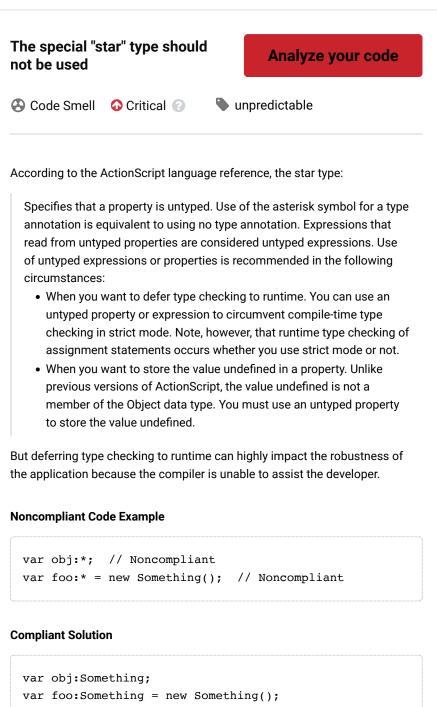Methods should not be empty

☢ Code Smell

Constant names should comply with a naming convention

☢ Code Smell

All branches in a conditional structure should not have exactly the same implementation

🐛 Bug

Classes that extend "Event" should

The special "star" type should not be used

☢ Code Smell | ⬆ Critical ⦻ | 🏷 unpredictable

Analyze your code

According to the ActionScript language reference, the star type:

> Specifies that a property is untyped. Use of the asterisk symbol for a type annotation is equivalent to using no type annotation. Expressions that read from untyped properties are considered untyped expressions. Use of untyped expressions or properties is recommended in the following circumstances:
> - When you want to defer type checking to runtime. You can use an untyped property or expression to circumvent compile-time type checking in strict mode. Note, however, that runtime type checking of assignment statements occurs whether you use strict mode or not.
> - When you want to store the value undefined in a property. Unlike previous versions of ActionScript, the value undefined is not a member of the Object data type. You must use an untyped property to store the value undefined.

But deferring type checking to runtime can highly impact the robustness of the application because the compiler is unable to assist the developer.

Noncompliant Code Example

```
var obj:*;  // Noncompliant
var foo:* = new Something();  // Noncompliant
```

Compliant Solution

```
var obj:Something;
var foo:Something = new Something();
```

Available In:

sonarcloud ☁ | sonarqube ⌇

override "Event.clone()"

🐞 Bug

Constructors should not dispatch
events

🐞 Bug

"ManagedEvents" tags should have
companion "Event" tags

🐞 Bug

Objects should not be instantiated
inside a loop

☢ Code Smell

Two branches in a conditional
structure should not have exactly the
same implementation

☢ Code Smell

Constructor bodies should be as
lightweight as possible

☢ Code Smell

Only "while", "do" and "for" statements
should be labelled

☢ Code Smell

Statements, operators and keywords
specific to ActionScript 2 should not
be used

☢ Code Smell

"for" loop stop conditions should be
invariant

☢ Code Smell

Unused function parameters should
be removed

☢ Code Smell